Fundamentos de bases de datos
Tarea 3
Modelo Relacional

Díaz Gómez Silvia
Eugenio Aceves Narciso Isaac
Quiroz Castañeda Edgar

22 de marzo del 2019

1. Preguntas de repaso

a. ¿Qué es una **relación** y qué características tiene?

R: Una relación es una representación de las interacciones que puede haber entre entidades que nos interesan dentro de la *base de datos*. Está definida como su homologo en matemáticas (Subconjunto del producto cartesiano...) considerando a las entidades o más especificamente a sus atributos como las estructuras entre las que se define.
Las relaciones en el modelo E/R se caracterizan por:

a) **Grado** El número de entidades que involucran (no necesariamente deben ser distintas).
Lineas que salen ó entran a la relación.

b) **Patricipación** Expresa si esta interacción ocurre con todos los ejemplares de la entidad involucrada o no.
doble linea o linea simple.

c) **Cardinalidad** La 'proporción' entre ejemplares de las entidades relacionadas, *Uno* ó *Muchos*. Por ejemplo, Un Banco muchas Cuentas ó Un Banco un Gerente ó Muchas tareas muchos alumnos.
Agregar una flecha al final de la linea tocando la entidad (Uno).

d) **Identificación** Cuando una relación conecta una entidad débil con la entidad fuerte que ayuda a la débil a identificarse usamos esto para denotar esta meta-relación.
Doble rombo.

e) **Atributos** En nuestros diesños podemos agregar atributos a las Relaciones como si fueran entidades. Esto cambia algunas cosas cuando se traduce y es preferible usar una entidad asociativa si se tienen muchos atributos.
No hay llaves para Relaciones pero pueden ser todo lo demás.
Óvalos conectados.

Aparte de esto, *relación* también es el nombre que se le da a una tabla en la Base de datos, así como se dice que los renglones son tuplas y las columnas miembros/dominios.

b. ¿Qué es un **esquema de relación**?

R: ¿Esto se refiere al esquema relacional? De ser así, es una notación por 'Tuplas' que sirve para representar las tablas que se obtienen a partir de un diagrama E/R. Estas tuplas son de la forma .$^{\text{En}}$tidad(Llave,LlaveForánea,Atributo,...)ƨ para generarlas se siguen algunas regals de traducción.

c. ¿Qué es una **llave primaria** ¿qué es una **llave candidata**? ¿qué es una **llave mínima**? ¿qué es una **super llave**?

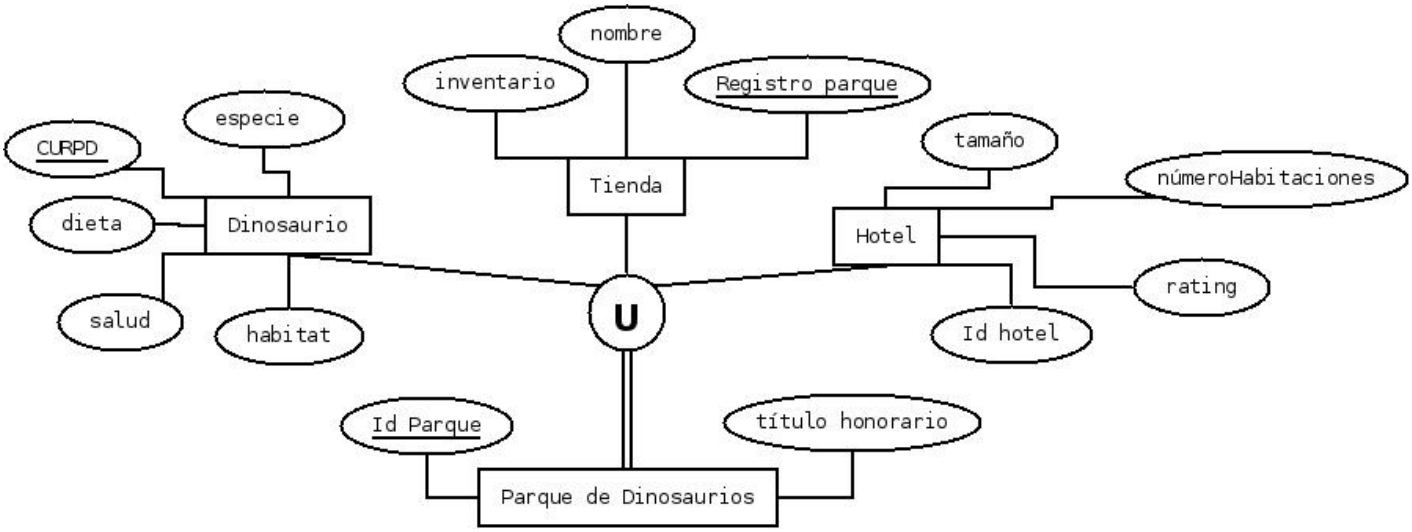
R: Una llave primaria es un atributo o conjunto de atributos (por lo general mínimo) elegido para ser usado como medio para identificar de manera única a cada ejemplar de una relación (tabla).
Una llave candidata es una posible llave, es decir, un conjunto de atributos que podrían identificar a cada ejemplar de manera única. No necesariamente es la llave primaria, sea por alguna razón ó solamente como elección de diseño. Además, toda llave candidata garantiza que ningún subconjunto dentro de ella es llave candidata.
Una llave mínima es la llave candidata de una relación con la propiedad de tener el menor número de elementos (ser mínimo, no minimal).
Finalmente una superllave es similar a llave candidata, pero suprimiendo la condición de no contener llaves. Es decir, cualquier cosa mientras garantice que podría ser un identificador único.

d. ¿Qué restricciones impone una **llave primaria** y una llave foránea al modelo de dato relacional?

R: En implementación, ambas son causa de restricción por las políticas de mantenimiento e integridad referencial cuando se quiere hacer modificaciones. En el modelo relacional, una entidad no puede tener más de una llave primaria y esta llave primaria no debe ser usada por ninguna otra relación en la BD. Las llaves foráneas son como monedas de intercambio que se usan para que relaciones distintas puedan reconstruir vínculos o asociarse con ejemplares de otras relaciones. Puede haber varias llaves foráneas en una relación, todas perteneciendo a distintas relaciones. Siempre se utiliza la llave primaria (o lo más cercano) como llave foránea cuando es necesario pasarselo a otra relación.

e. Investiga cómo se traducen las **categorías** (presentes en el **modelos E/R**) al **modelos relacional**. Proporciona un ejemplo.

R: El ejemplo es el sigueinte diagrama.



Al traducir tendriamos:

- Parque_de_Dinosaurios(**Id Parque**, título honorario)
- Dinosaurio(**CURPD**,habitat,dieta,salud,especie,***Id Parque***)
- Hotel(**Id Hotel**,tamaño,rating,númeroHabitaciones,***Id Parque***)
- Tienda(**Registro Parque**,inventario,nombre,***Id Parque***)

2. Modelo relacional

Traduce el siguiente modelo **Entidad/Relación** a su correspondiente **Modelo Relacional**:

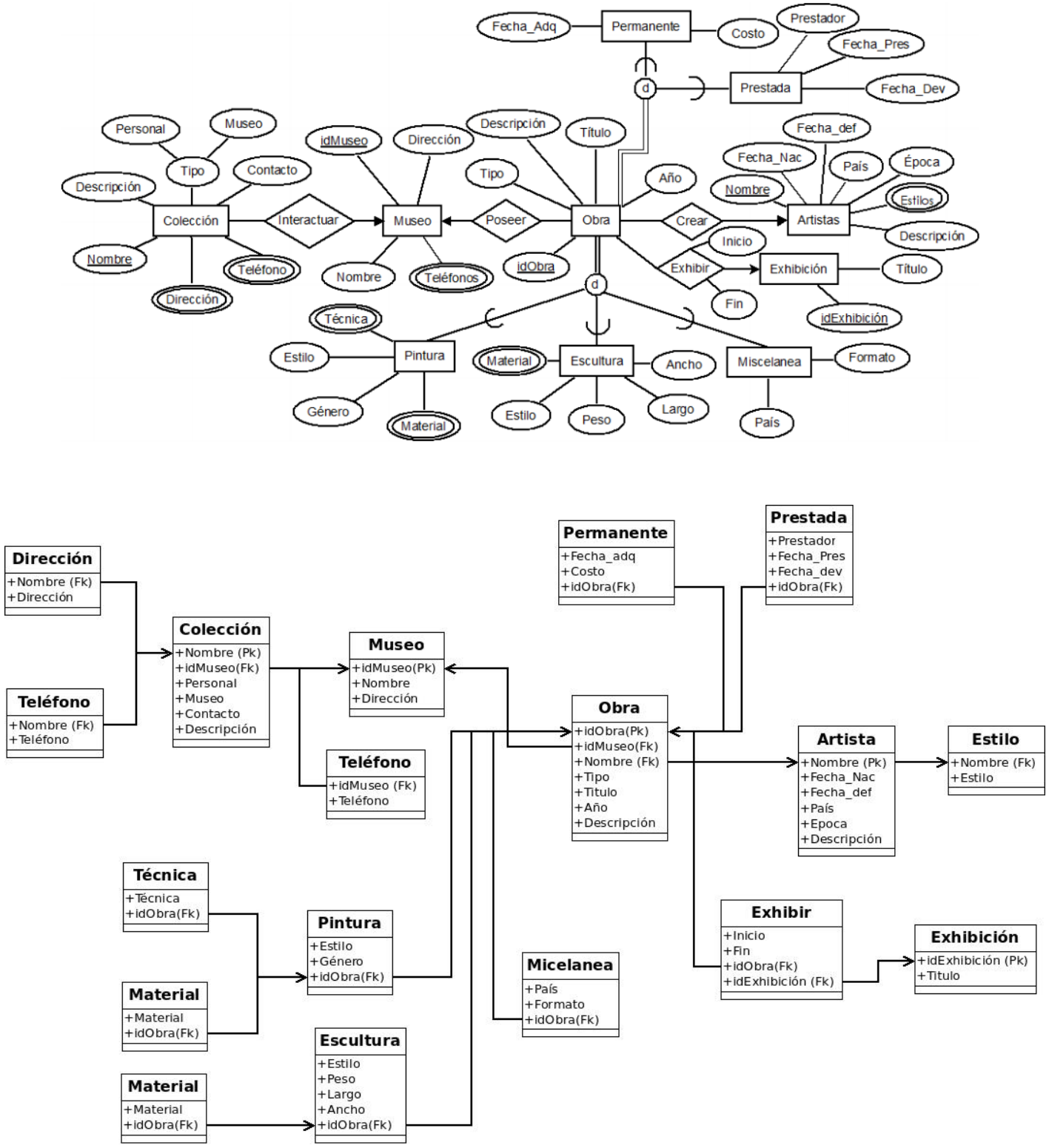


Figura 1: Traducción del modelo E-R de la figura anterior al modelo Relacional.

3. Modelo relacional

Traduce s su correspondiente **Modelo Relacional** el problema del **Sistema de Información Geográfica (Tarea 1)**. Se realizaste alguna modificación a tu diseño orignal (para mejorarlo) indica los cambios hechos y la justificación de los mismos.
En cualquier caso, deberás mostrar el **diagrama E/R** y su correspondiente traducción. Es importante que muetres tanto las **restricciones de entidad** como las de **integridad referencial**.

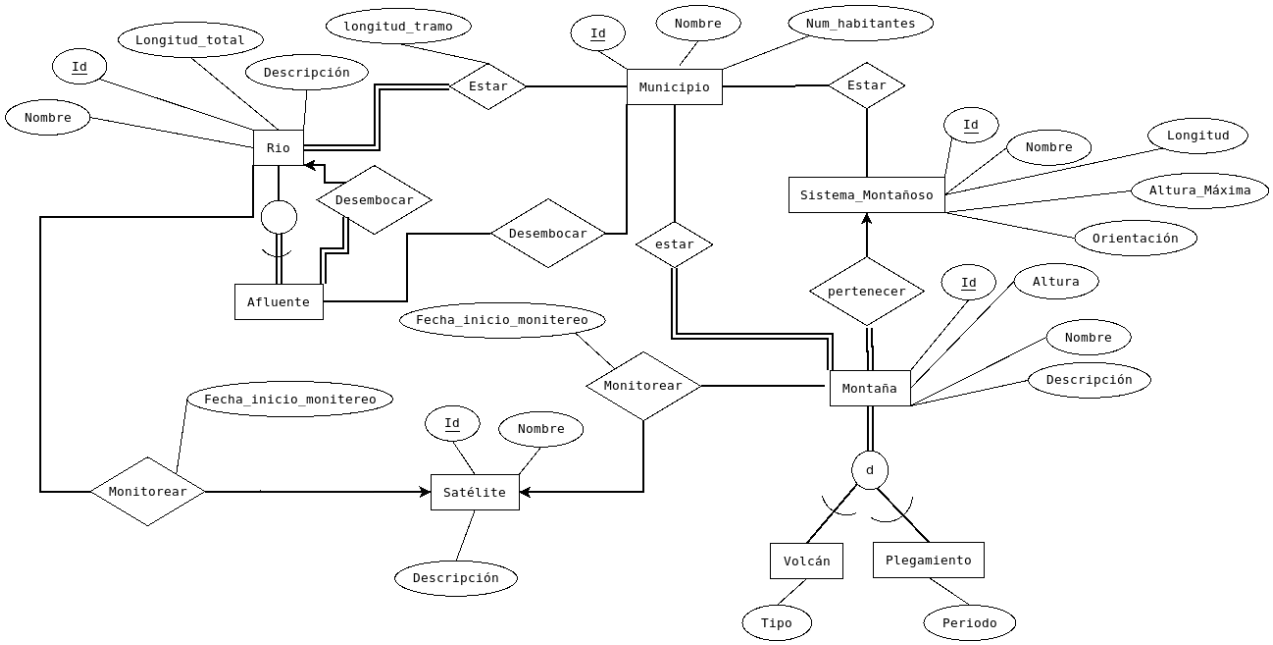


Figura 2: Modelo E-R para el Sistema de Información Geográfica.

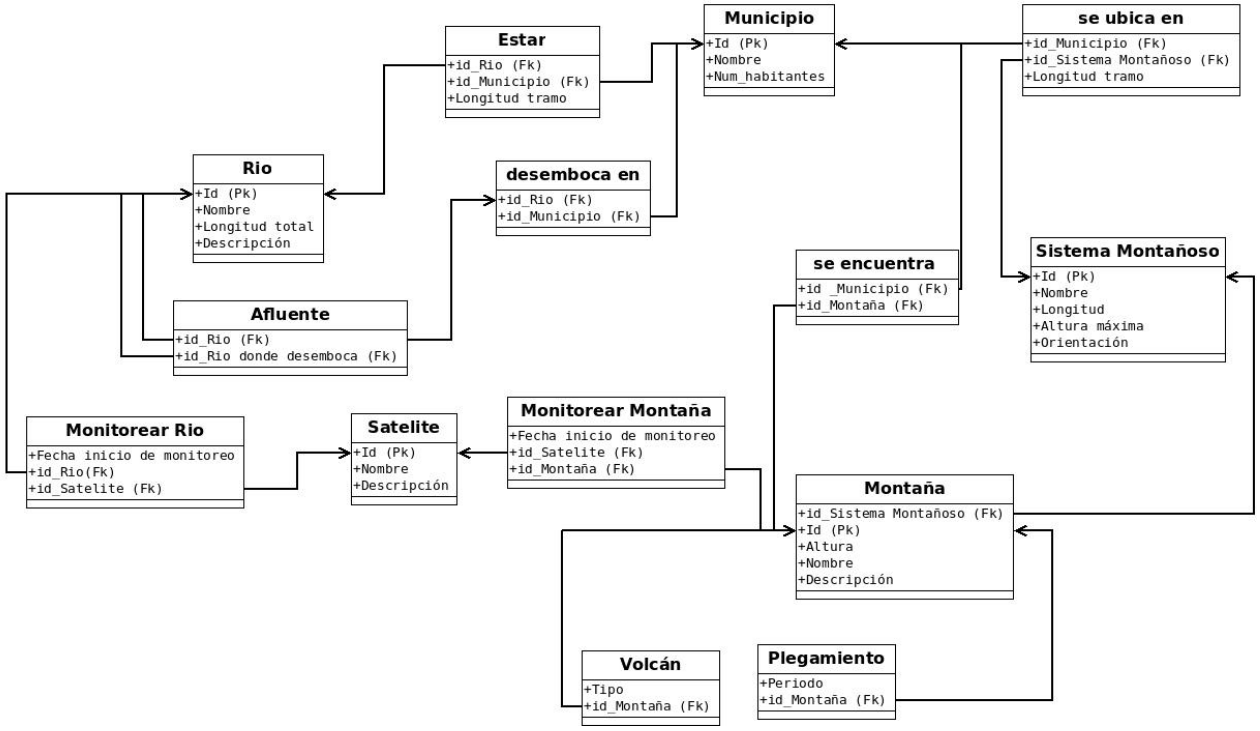


Figura 3: Traducción del modelo E-R para el Sistema de Información Geográfica a Esquema Relacional.

4. Lectura

Leer el artícula **Codd's 12 Rules for a RDBMS**. Explica con tus propias palabras cada una de las 12 reglas de **Codd**.
Indica por qué consideras que son importantes y si, hasta el momento de lo comentado en el curso, sería posible que un **SMBD** pudiera cumplir enteramente con lo que ahí se propone.

- **Regla 1 : The Information Rule**
  Toda la información debe estar representada en el esquema lógico de la base de datos. Y debe estar represetada como entradas de una tabla.
  Si los datos no están en el esquema lógico, entonces no sería accesibles usando los mecanismos de consulta del sistema, por lo que es como si los datos no estuvieran ahí.
  Esto se puede lograr en SMBD forzando a que la única manera de almacenar datos sea a través de inserciones en tablas.

- **Regla 2 : Guaranteed Access Rule** Toda la información debe ser accesible. Esto es que todo dato tenga un identificador o llave primaria, además de que los datos sean atomicos ya que son de suma importancia para poder garantizar la accesibilidad a los datos.
  Esto es importante porque no poder acceder a los datos es equivalente a no tener los datos.
  Sería posible que un SMBD pueda garantizar esto si a todos los datos insertdos bien se les exige una llave al momento o se les asigna un llave sintética al ser ingresados.
  Aunque forzar lo segundo podría llevar bien a redundancia o que ya no se refleje totalmente el esquema lógico de la base de datos.

- **Regla 3: Systematic Treatment of NULL Values**
  Debe existir un valor NULL diferente de todos los demás posibles datos que represente la ausencia de información y éste debe ser tratado de la misma manera que todos los demás datos por el sistema.
  Esto es importante porque en otro caso no se podría represetar la ausencia de información.
  Esto se puede lograr en un SMBD simplemente añadiendo el caso.

- **Regla 4: Dynamic Online Catalog Based on the Relational Model**
  La estructura de la base de dato debe estar almacenada de igual manera que el resto de los datos, para que pueda ser accesidad por usuarios usando los mecanismos ordinarios de consulta.

- **Regla 5: Comprehensive Data Sublanguage Rule**
  Se debe tener soporte para algún lenguaje formal que permita definir tipos de datos, hacer consultas, transacciones, y definir restricciones.

- **Regla 6: View Updating Rule**
  Todas las vistas que son teóricamente actualizables deben de poder ser actualizadas.

- **Regla 7: High-Level Insert, Update, and Delete**
  Las operaciones de modificación de la base de datos deben estar disponilbes en términos de conjuntos de tuplas, no de tuplas individuales.

- **Regla 8: Physical Data Independence**
  Esta regla menciona que a nivel fisico que es donde la base de datos almacena e implementa los metodos de acceso a los datos es independiente de la manera lógica en que se accede, por lo tanto los cambios que se hagan a nivel físico no le afecta al usuario puesto que al usuario no le interesa saber como se almacena o como se acceden a los datos.

- **Regla 9: Logical Data Independence**
  El esquema de la base de datos debe ser independiente de las aplicaciones que utilizan la base de datos. Esto es que se pueda modificar la estrucutura lógica de la base de datos sin necesidad de modificar las aplicaciones que la utilizan.

- **Regla 10: Integrity Independence**
  Las restricciones de integridad deben de ser independientes del funcionamiento de aplicaciones. Esto es que esas restricciones se puedan modificar sin necesidad de modificar las aplicaciones que utilizan la base de datos.

- **Regla 11: Distribution Independence**
  Una base de datos distribuida debe ser idéntica a una no distribuida desde el punto de vista del usuario.

- **Regla 12: Non-Subversion Rule**
  Indica que no debe existir un mecasnimo para violar las restricciones. Es decir, en caso de que se propocione acceso de bajo nivel al sistema, esta interfaz no debe ser capaz de modificar sin restricciones.